

TEMPORIZADORES (TIMERS) DO dsPIC30F4011

INTRODUÇÃO

- ❑ Dependendo da variante específica, a família de dispositivos dsPIC30F oferece vários temporizadores de 16 bits. Estes temporizadores tem nomes Timer1, Timer2, Timer3,..., etc.
- ❑ Cada módulo Timer é um Temporizador/Contador de 16 bits consistindo dos seguintes registradores de leitura/escrita: TMRx (Registrador de contagem do timer de 16 bits), PRx (Registrador de período de 16 bits associado com o timer), TxCON (Registrador de Controle de 16 bits associado com o timer).
- ❑ Cada módulo Timer também tem os bits associados para o controle de interrupção: TxIE (bit de controle de habilitação de interrupção), TxIF (bit de estado do flag de interrupção) e TxIP<2:0> (bits de controle de Prioridade de Interrupção).
- ❑ Com certas exceções, todos os timers de 16 bits tem o mesmo circuito funcional. Alguns timers podem ser combinados para formar um timer de 32 bits.

TEMPORIZADORES DO dsPIC30F4011

MODULO TEMPORIZADOR 1 (TIMER1)

- ❑ Este é um temporizador de 16 bits o qual pode servir como contador de tempo para Clock de Tempo Real (Real time Clock – RTC), ou operar como um temporizador/contador free – running.
- ❑ O temporizador de 16 bits tem os seguintes modos: **Temporizador de 16 bits, Contador Síncrono de 16 bits, Contador Assíncrono de 16 bits.**
- ❑ As seguintes características operacionais são suportadas: **operação de timer gatilhado, configuração de prescaler selecionável, operação de temporização durante os modos Idle, e Sleep da CPU e interrupções sobre o registrador de Período ou na borda de descida do sinal de porta externa.**
- ❑ Estes modos de operação são determinados configurando os bits apropriados do registrador T1CON.

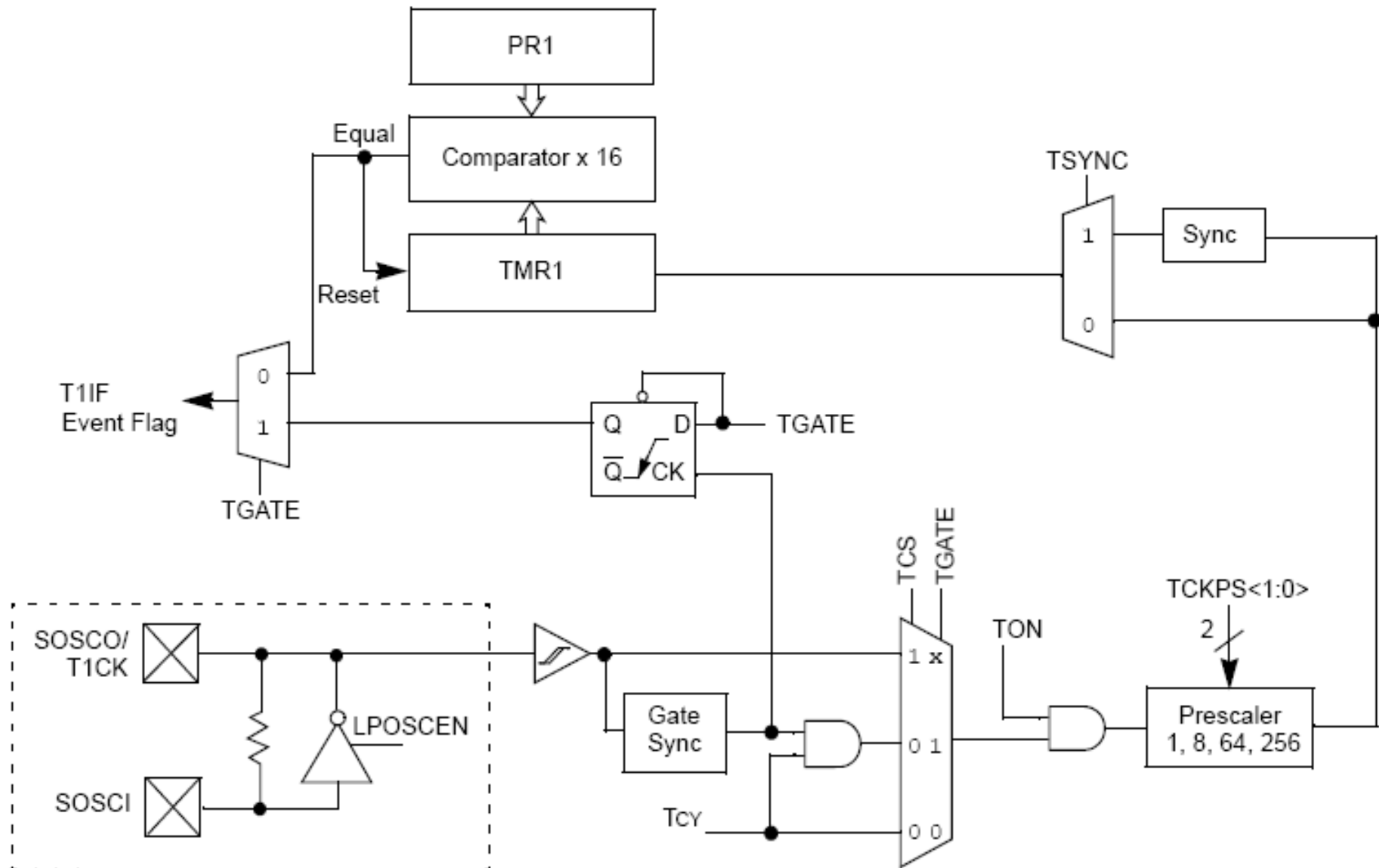
TIMER1 REGISTER MAP⁽¹⁾

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
TMR1	0100	Timer1 Register																uuuu uuuu uuuu uuuu
PR1	0102	Period Register 1																1111 1111 1111 1111
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000 0000 0000 0000

Legend: u = uninitialized bit; — = unimplemented bit, read as "0"

Note 1: Refer to the "dsPIC30F Family Reference Manual" (DS70048) for descriptions of register bit fields.

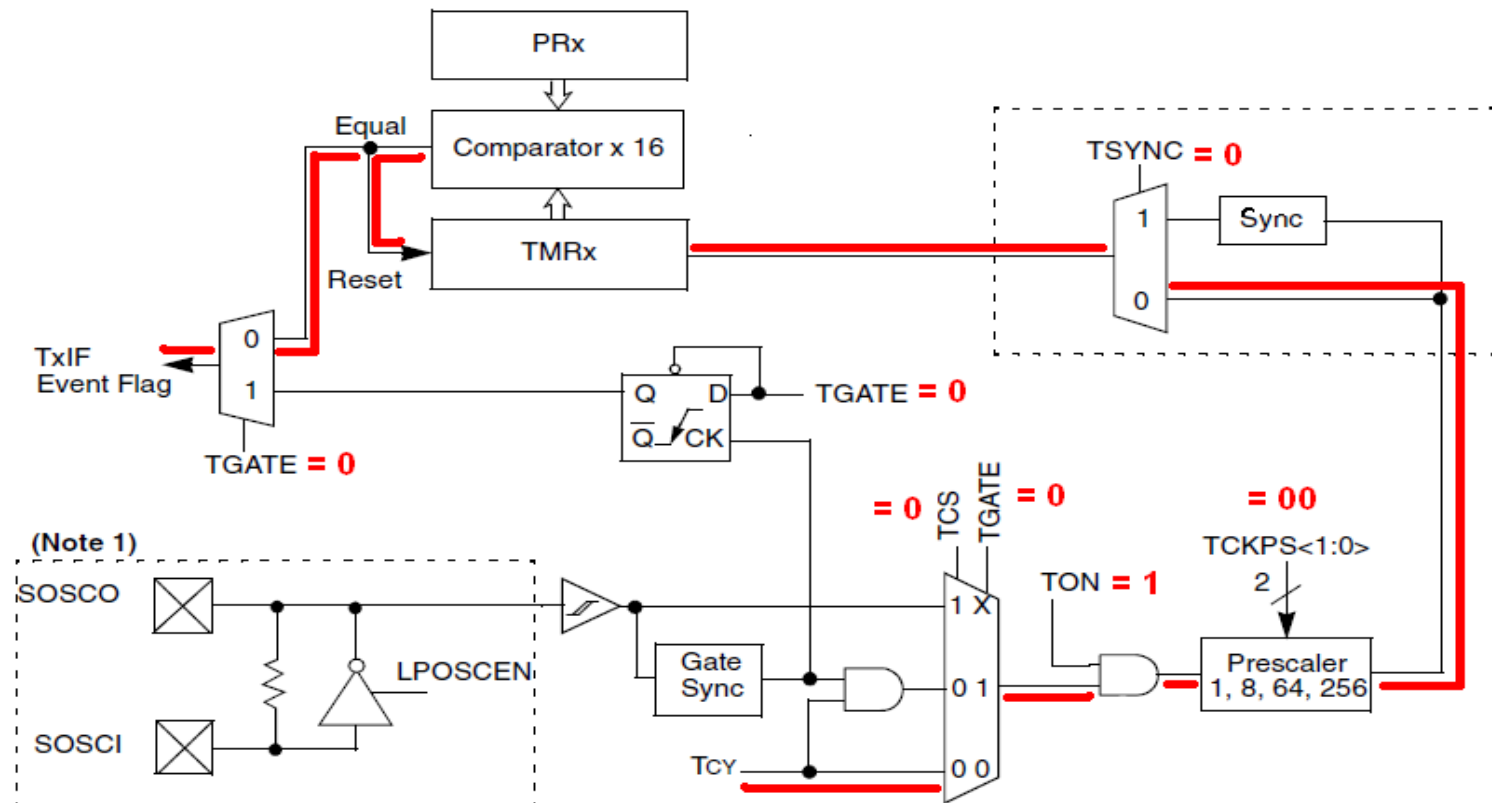
16-BIT TIMER1 MODULE BLOCK DIAGRAM (TYPE A TIMER)



TICON : Timer1 Control Register (Tipo A)					
Bit	Nome	Descrição	Bit	Nome	Descrição
0	-	Não Implementado. Ler como 0	6	TGATE	Timer Gated Time Accumulation Enable bit 1 = Gated time accumulation enabled 0 = Gated time accumulation disabled (TCS must be set to '0' when TGATE = 1. Reads as '0' if TCS = 1)
1	TCS	Timer Clock Source Select bit 1 = External clock from pin TxCK 0 = Internal clock (FOSC/4)	7-12	-	Não Implementado. Ler como 0
2	TSYNC	Timer External Clock Input Synchronization Select bit <u>When TCS = 1:</u> 1 = Synchronize external clock input 0 = Do not synchronize external clock input <u>When TCS = 0:</u> This bit is ignored. Read as '0'. Timer1 uses the internal clock when TCS = 0	13	TSIDL	Stop in Idle Mode bit 1 = Discontinue timer operation when device enters Idle mode 0 = Continue timer operation in Idle mode
3	-	Não Implementado. Ler como 0	14	-	Não Implementado. Ler como 0
5-4	TCKPS<1:0>	Timer Input Clock Prescale Select bits 11 = 1:256 prescale value 10 = 1:64 prescale value 01 = 1:8 prescale value 00 = 1:1 prescale value	15	TON	Timer On Control bit 1 = Starts the timer 0 = Stops the timer

MODO TEMPORIZADOR DE 16 BITS

- ❑ Todos os tipos de temporizadores tem a habilidade de operar no modo Temporizador(Timer).
- ❑ No modo Timer, o clock de entrada ao temporizador é fornecido pelo clock do sistema interno (FOSC/4).
- ❑ Quando habilitado, o temporizador incrementa uma vez por ciclo de instrução com uma configuração de prescaler de 1:1.
- ❑ O modo Timer é selecionado zerando o bit de controle TCS (TxCON<1>). O bit de controle de modo síncrono, TSYNC (T1CON <2>), não tem efeito, desde que a fonte de clock do sistema é usado para gerar o clock do timer.



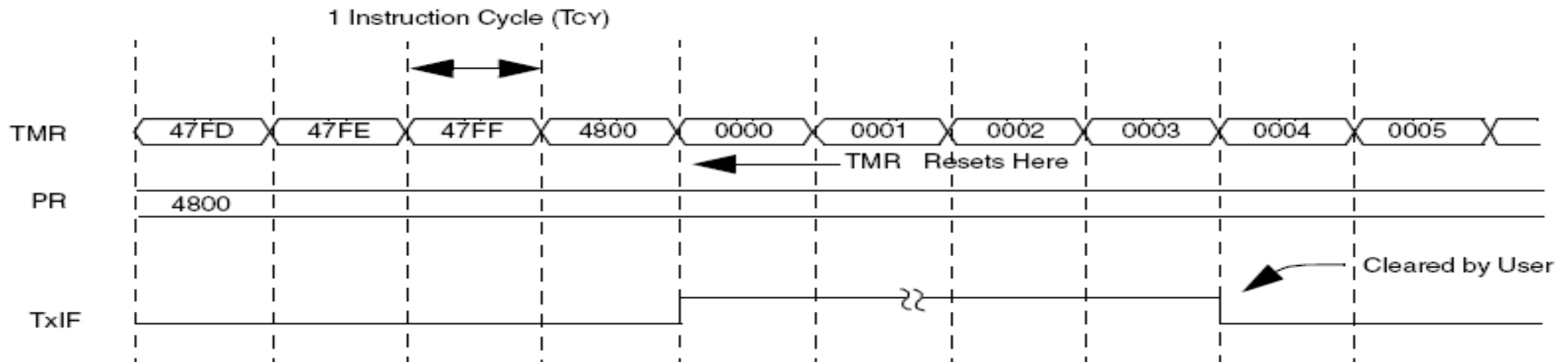
MODO TEMPORIZADOR DE 16 BITS

- ❑ O temporizador incrementa em cada **ciclo de instrução (Tcy)** (quatro vezes o período do relógio de entrada - XTAL) até que case com o valor pré-carregado no registrador de período, PR1, então reseta a 0 e continua a conta, nesse momento uma requisição de interrupção pelo módulo timer1 é gerada (T1IF) no registrador IFS0. O processamento desta requisição depende do bit de habilitação de interrupção T1IE (no registrador IEC0). O módulo timer continua a operar durante a rotina de interrupção.
- ❑ Quando a CPU vai para o modo IDLE, o timer pára de incrementar a menos que o bit TSIDL igual a 0 (T1CON<13>). Se TSIDL = 1, a lógica do timer incrementará a seqüência até o termino do modo IDLE da CPU.

❑ O tempo do ciclo de execução de uma instrução => $T_{cy} = 1/\text{MIPS}$.

❑ Frequência de execução de uma instrução => $\text{MIPS} = (\text{Fosc} * \text{PLLx}) / 4$.

Interrupt Timing for Timer Period Match



```
//Criar uma função para ter atrasos na ordem dos milisegundos.
#include <p30f4011.h>
//=====
_FOSC(CSW_FSCM_OFF & XT); //A frequencia do XTAL é de 8 MHZ no kit, não utilizamos PLL.
_FWDT(WDT_OFF);
_FBORPOR(PBOR_ON & PWRT_OFF & MCLR_EN);
//=====
//*****Função Delay em ms *****
Delay_ms(unsigned int tempo)
{
    while (tempo != 0)
    {
        while(IFS0bits.T1IF==0); //enquanto este bit for 0 fica nessa instrução (1 ms)
        IFS0bits.T1IF=0;
        tempo--;
    }
}

//***** Programa Principal *****
void main (void)
{
    ADPCFG = 0xFFFF; //configura a porta B (PORTB)
                        // como entradas/saídas digitais
    TRISB=0;          //a PORTB como saída
    PR1 = 2000;        //O registrador de periodo PR1 é igual a 2000 pelos seguintes motivos:
                        // 1) MIPS = (Fosc * PLLx)/4 = (8 MHz * 1)/4 = 2 MIPS
                        // Tcy = 1/MIPS = 1 / 2 = 0,5 useg
                        // 2) Para dar 1 mseg temos que multiplicar Tcy vezes 2000.
    T1CON=0x8000; //ativamos o timer1 e o Prescaler fica em 1.
    Delay_ms(500);
    ....
}
```

//Escrever um programa usando o Timer 1 para criar uma onda quadrada de 10 KHz no pino 0 da porta B. Sabendo que a
 // frequência do cristal é de 8MHz, se utiliza o PLL x 8 e o prescaler do timer é 1.

//Solução:

//como precisamos uma saída de 10 KHz o período deste sinal será de $T = 100 \text{ useg}$. Em ALTO 50 useg e em BAIXO outros 50 useg.

//Portanto, temos que ter um delay de 50 useg, para isso devemos fazer os seguintes cálculos:

// $T_{cy} = 1/\text{MIPS}$; $\text{MIPS} = (\text{Fosc} * \text{PLLx})/4$; $T(50) = \text{tempo de } 50 \text{ useg}$; $T(50) = T_{cy} * \text{PR1}$. Devemos encontrar o valor de PR1

//Então $\text{PR1} = T(50)/T_{cy} = T(50) * (\text{Fosc} * \text{PLLx})/4 = (50 \text{ useg}) * (8 \text{ MHz} * 8)/4 = 800$

//*****

#include <p30f4011.h>

//=====

_FOSC(CSW_FSCM_OFF & XT_PLL8); //A frequencia do XTAL é de 8 MHZ no kit, utilizamos PLL8.

_FWDT(WDT_OFF);

_FBORPOR(PBOR_ON & PWRT_OFF & MCLR_EN);

//=====

//***** Programa Principal *****

void main (void)

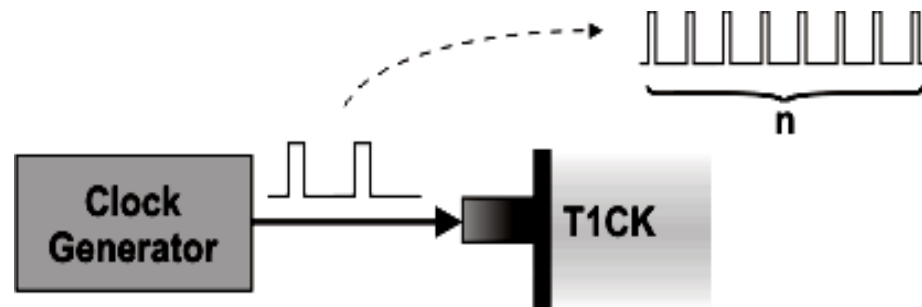
```
{
    ADPCFG = 0xFFFF; //configura a porta B (PORTB) como entradas/saídas digitais
    TRISB=0;          //a PORTB como saída
    IFS0=0;           //Flag de interrupção do timer1
    LATB=0;
    PR1 = 800; //O registrador de periodo PR1 é igual a 800
    T1CON=0x8000; //ativamos o timer1 e o Prescaler fica em 1.
    while(1)
    {
        while (IFS0bits.T1IF ==0); //Fica nessa instrução até que esse bit mude para 1
        IFS0bits.T1IF = 0;
        LATBbits.LATB0 = ~LATBbits.LATB0; //Complementamos o bit.
    }
}
```



```
//Escrever um programa usando o Timer 1 para criar uma onda quadrada de 10 KHz no pino 0 da porta B. Sabendo que a
// frequência do cristal é de 8MHz, se utiliza o PLL x 8 e o prescaler do timer é 1. Usar interrupções (Timer1).
//*****
#include <p30f4011.h>
//=====
_FOSC( CSW_FSCM_OFF & XT_PLL8);          //A frequencia do XTAL é de 8 MHZ no kit, utilizamos PLL8.
_FWDT(WDT_OFF);
_FBORPOR(PBOR_ON & PWRT_OFF & MCLR_EN);
//=====
void __attribute__((interrupt, no_auto_psv)) _T1Interrupt(void)
{
    IFS0bits.T1IF = 0;
    LATBbits.LATB0 = ~LATBbits.LATB0; //Complementamos o bit.
}
//***** Programa Principal *****
void main (void)
{
    ADPCFG = 0xFFFF;    //configura a porta B (PORTB) como entradas/saídas digitais
    TRISB=0;             //a PORTB como saída
    IFS0=0;               //Flag de interrupção do timer1
    LATB=0;
    IEC0 = IEC0 | 0x0008; // bit 3 do registrador IEC0 habilita a interrupção do timer1 (IEC0bits.T1IE=1)
    PR1 = 800; //O registrador de periodo PR1 é igual a 800
    T1CON=0x8000; //ativamos o timer1 e o Prescaler fica em 1.
    while(1);    //laço infinito
}
```

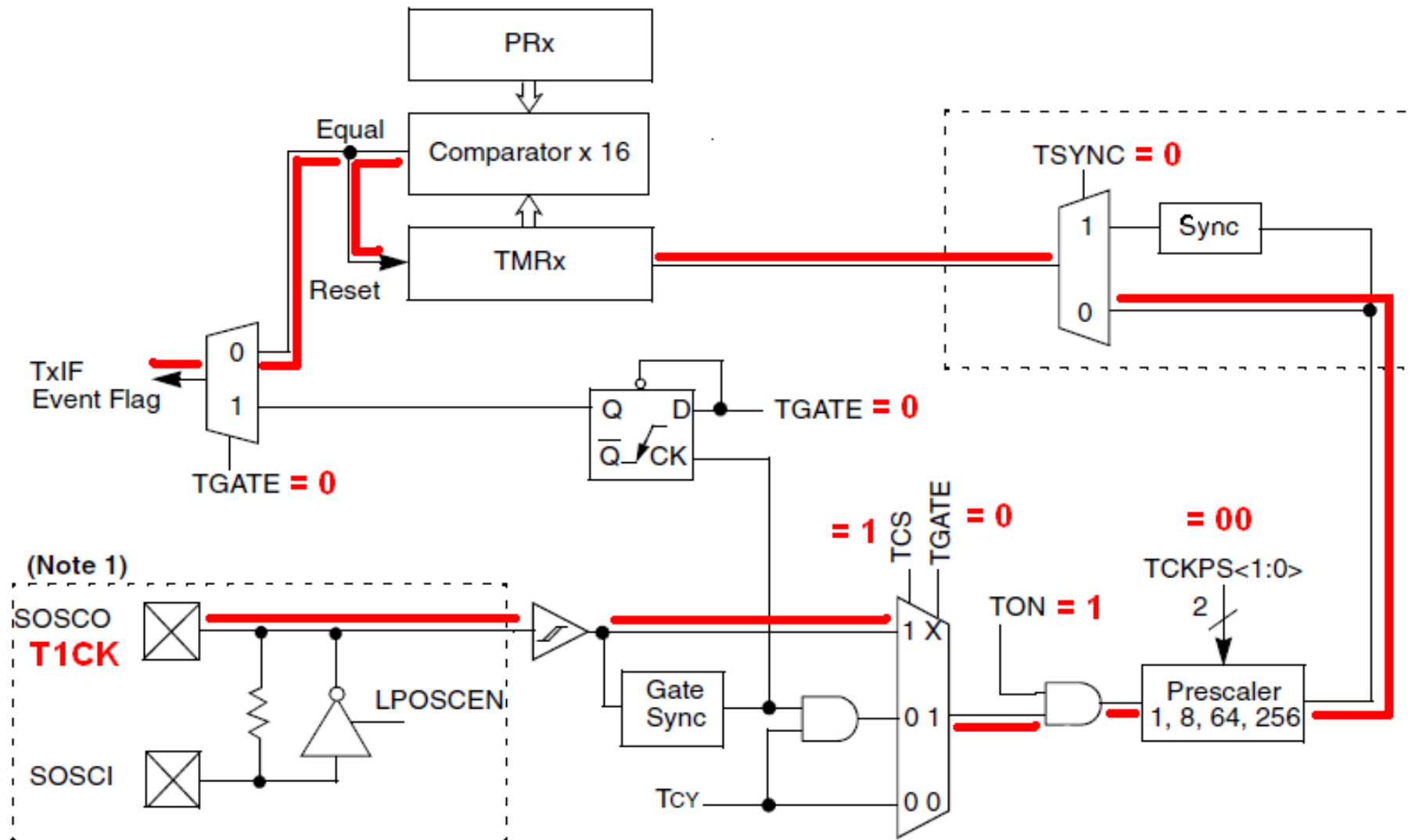

MODO CONTADOR SÍNCRONO DE 16 BITS

- ❑ O temporizador incrementa (TMR1) na borda de subida do sinal de clock externo aplicado, o qual é sincronizado com a fase interna do clock(o clock de entrada externo está sempre sincronizado ao clock de ciclo de instrução do sistema, T_{cy}). O timer conta até que case com o valor pré-carregado em PR1, então reseta para 0 e continua, nesse momento uma requisição de interrupção pelo módulo timer1 é gerada (T1IF) no registrador IFS0. O processamento desta requisição depende do bit de habilitação de interrupção T1IE (no registrador IEC0). O módulo timer continua a operar durante a rotina de interrupção.
- ❑ Quando a CPU vai para o modo IDLE, o timer para de incrementar a menos que o bit TSIDL igual a 0 ($T1CON<13>$). Se $TSIDL = 1$, a lógica do timer incrementará a sequencia até o termino do modo IDLE da CPU.
- ❑ O propósito do timer1 é permitir medições de períodos de sinais de clock muito rápidos, por exemplo, autodetecção da velocidade de comunicação da interface serial universal UART.
- ❑ **Exemplo:** Incrementar o valor da porta B a cada décimo pulso. Neste exemplo o timer1 é usado para contar pulsos de clock externos no pino T1CK. Depois de dez pulsos acontece a interrupção no timer1 e o valor da porta B é incrementada.



```
#include <p30f4011.h>
//=====
_FOSC( CSW_FSCM_OFF & XT_PLL8);          //A frequencia do XTAL é de 8 MHZ no kit, utilizamos PLL8.
_FWDT(WDT_OFF);
_FBORPOR(PBOR_ON & PWRT_OFF & MCLR_EN);
//=====
void __attribute__((interrupt, no_auto_psv)) _T1Interrupt(void)
{
    IFS0bits.T1IF = 0;
    LATB++; //incrementamos o valor da porta B.
}
//***** Programa Principal *****/
void main (void)
{
    ADPCFG = 0xFFFF;    //configura a porta B (PORTB) como entradas/saídas digitais
    TRISB=0;             //a PORTB como saída
    TRISC = 0x4000;      // PORTC<14>=1; pino T1CK de entrada
    IFS0=0;              //Flag de interrupção do timer1
    LATB=0;
    IEC0 = IEC0 | 0x0008; // bit 3 do registrador IEC0 habilita a interrupção do timer1 (IEC0bits.T1IE=1)
    PR1 = 10;            //O registrador de periodo PR1 é igual a 10
    T1CON=0x8006;        //ativamos o timer1 como contador síncrono, o Prescaler fica em 1, ativamos TSYNC e TCS(T1CK)
    while(1);            //laço infinito
}
```

MODO CONTADOR ASÍNCRONO DE 16 BITS

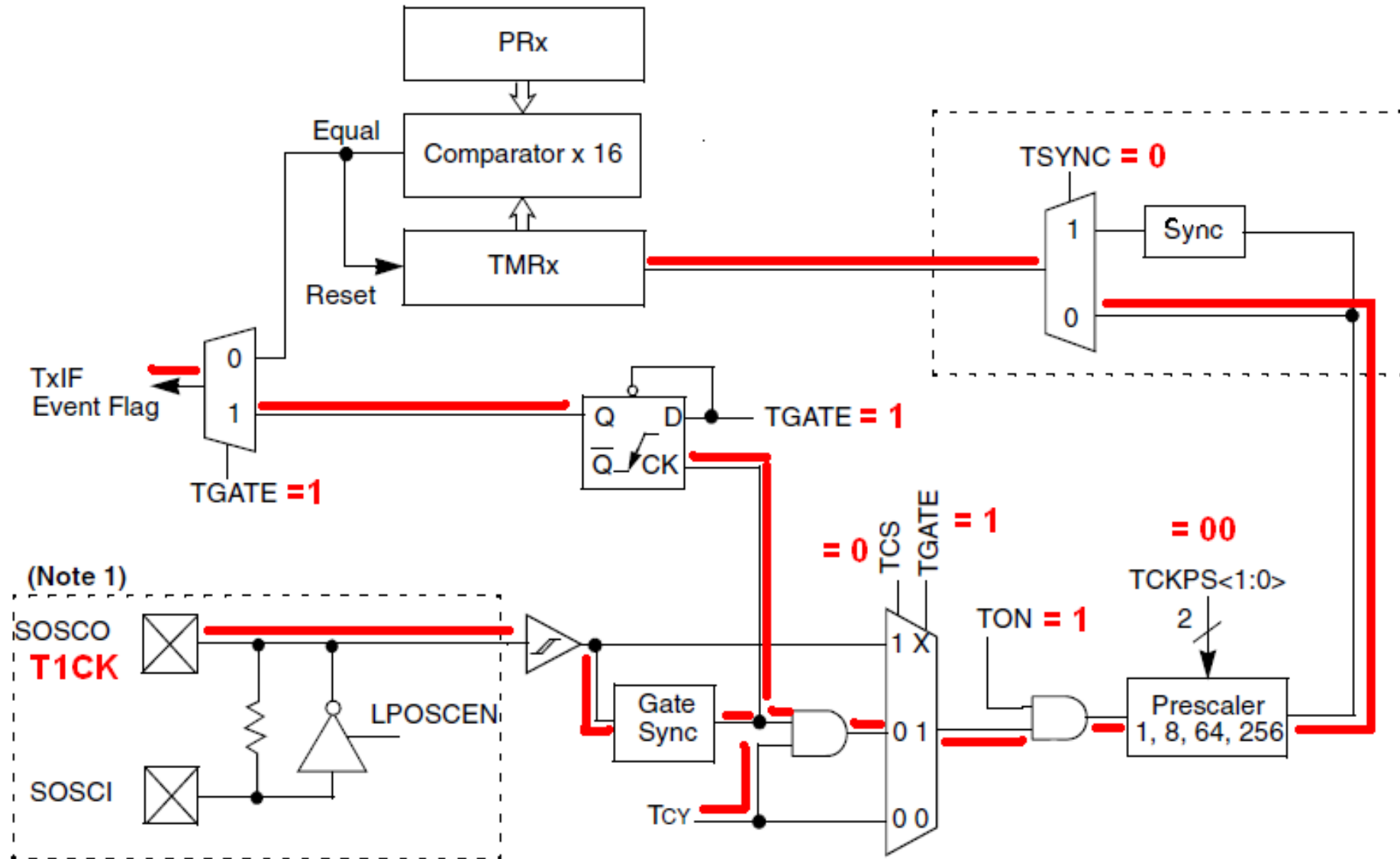


MODO CONTADOR ASSÍNCRONO DE 16 BITS

- ❑ O temporizador incrementa (TMR1) na borda de subida do sinal de clock externo aplicado, o qual **não** é sincronizado com a fase interna do clock. O timer conta até que case com o valor pré-carregado em PR1, então reseta para 0 e continua, nesse momento uma requisição de interrupção pelo módulo timer1 é gerada (T1IF) no registrador IFS0. O processamento desta requisição depende do bit de habilitação de interrupção T1IE (no registrador IEC0). O módulo timer continua a operar durante a rotina de interrupção.
- ❑ Quando a CPU vai para o modo IDLE, o timer pára de incrementar a menos que o bit TSIDL igual a 0 (T1CON<13>). Se TSIDL = 1, a lógica do timer incrementará a seqüência até o termino do modo IDLE da CPU.
- ❑ **Exemplo:** Contar cada 800 pulsos e incrementar o valor da porta B. Neste exemplo o timer1 é usado para contar a cada oitavo pulso de um clock externo no pino externo T1CK. Depois de 100 pulsos a rotina de interrupção é chamada e o valor da porta B é incrementada.

```
#include <p30f4011.h>
//=====
_FOSC( CSW_FSCM_OFF & XT_PLL8);          //A frequencia do XTAL é de 8 MHZ no kit, utilizamos PLL8.
_FWDT(WDT_OFF);
_FBORPOR(PBOR_ON & PWRT_OFF & MCLR_EN);
//=====
void __attribute__((interrupt, no_auto_psv)) _T1Interrupt(void)
{
    IFS0bits.T1IF = 0;
    LATB++; //incrementamos o valor da porta B.
}
//***** Programa Principal *****/
void main (void)
{
    ADPCFG = 0xFFFF;    //configura a porta B (PORTB) como entradas/saídas digitais
    TRISB=0;             //a PORTB como saída
    TRISC = 0x4000;      // PORTC<14>=1; pino T1CK de entrada
    IFS0=0;              //Flag de interrupção do timer1
    LATB=0;
    IEC0 = IEC0 | 0x0008; // bit 3 do registrador IEC0 habilita a interrupção do timer1 (IEC0bits.T1IE=1)
    PR1 = 100; //O registrador de período PR1 é igual a 100, quando TMR1 é igual a PR1 se ativa a interrupção
    T1CON=0x8012; //ativamos o timer1 como contador assíncrono, o Prescaler fica em 8 ou seja o TMR1
                  // contará cada 8 pulsos e ativamos TCS(T1CK)
    while(1); //laço infinito
}
```

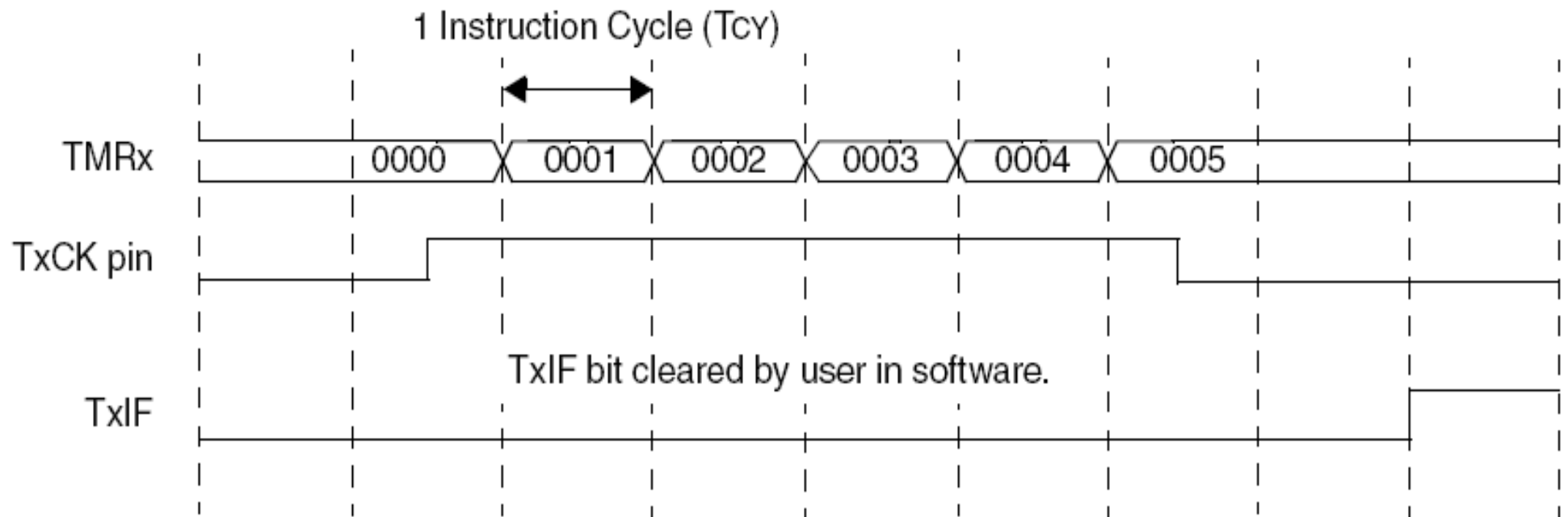
MODO GATED TIME ACCUMULATION



MOD0 GATED TIME ACCUMULATION

- ❑ Permite que o Tcy interno incremente o TMR1 quando o sinal da porta de entrada (pino T1CK) está em ALTO, assim poderia se medir a duração do tempo em ALTO. Quando o estado do pino TxCK é ALTO, o TMRx contará até que case com o registrador do período (PRx) ou quando o estado do pino é mudado para um estado BAIXO. **A transição do estado do pino desde alto para baixo setará o flag de interrupção TxIF.** O flag de interrupção vai para o nível ALTO depois de 1 ou 2 ciclos de instrução que aconteceu a borda de descida do sinal no pino TxCK.
- ❑ O bit de controle TGATE (TxCON<6>) deve ser setado para habilitar o modo Gated Time Accumulation. O timer deve ser habilitado, TON(TxCON<15>)=1, e a fonte de clock do timer setada para o clock interno TCS(TxCON<1>) = 0.
- ❑ A operação da gate começa na borda de subida do sinal aplicado ao pino TxCK e termina na borda de descida do sinal. O respectivo timer incrementará enquanto o sinal externo esteja em ALTO. A borda de descida do sinal termina a operação de contar, mas não reseta o timer. O usuário deve resetar o timer se este deseja começar de zero na próxima borda de subida do sinal. A borda de descida do sinal gera uma interrupção.
- ❑ A resolução da conta do timer está diretamente relacionada ao período do clock do timer. Para um prescaler de 1:1, o período do clock do timer é um ciclo de instrução. Para um prescaler de 1:256, o período do clock do timer é 256 vezes o ciclo de instrução. A resolução do clock do timer pode ser associado à largura de pulso do sinal gate.

Gated Timer Mode Operation



- ❑ **Exemplo:** Usar o timer1 no modo gated time accumulation. Habilitar GATE o sinal é aplicado ao pino T1CK. Medir a largura do nível ALTO do sinal e mostrar o resultado na porta B.

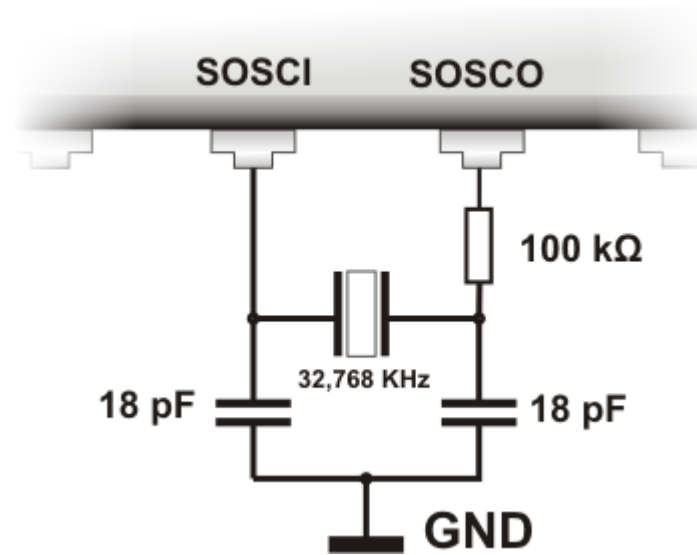
```
#include <p30f4011.h>
//=====
_FOSC( CSW_FSCM_OFF & XT_PLL8);          //A frequência do XTAL é de 8 MHZ no kit, utilizamos PLL8.
_FWDT(WDT_OFF);
_FBORPOR(PBOR_ON & PWRT_OFF & MCLR_EN);
//=====
void __attribute__((interrupt, no_auto_psv)) _T1Interrupt(void)
{
    IFS0bits.T1IF = 0;
    LATB = TMR1; //mostra-se na porta B a duração do pulso.
}
//***** Programa Principal *****/
void main (void)
{
    ADPCFG = 0xFFFF; //configura a porta B (PORTB) como entradas/saídas digitais
    TRISB=0;          //a PORTB como saída
    TRISC = 0x4000;    // PORTC<14>=1; pino T1CK de entrada
    IFS0=0;            //Flag de interrupção do timer1
    LATB=0;
    IEC0 = IEC0 | 0x0008; // bit 3 do registrador IEC0 habilita a interrupção do timer1 (IEC0bits.T1IE=1)
    PR1 =0xFFFF;        //O registrador de período PR1 é igual a FFFF
    T1CON=0x8040; //ativamos o timer1 como contador síncrono, ativamos TCS(T1CK) e TGATE
    while(1); //laço infinito
}
```

O valor de PR1 é grande devido a que permite medir intervalos de tempo tão grandes como seja possível.

MODO DE OPERAÇÃO CLOCK DE TEMPO REAL (REAL TIME CLOCK - RTC)

- ❑ O modulo timer1 pode ser ajustado para operar no modo de operação de clock de tempo real. Desta maneira obtemos a informação dos instantes de tempo (hora, minutos e segundos) servindo para a evidência de eventos. As principais características deste modo de operação são: uso do oscilador de 32 KHz, prescaler de 8 bits, baixa potência, e a habilidade de gerar uma requisição de interrupção RTC.
- ❑ Parecido a todos os modos de operação do timer1, este modo é configurado pelos bits de controle no registrador T1CON.
- ❑ Enquanto que o timer1 usa o clock do oscilador de 32KHz para operação no modo RTC, o resto do microcontrolador tem a habilidade de operar com outro clock o qual é ajustável pelos bits de controle no registrador de controle FOSC.
- ❑ Neste modo, quando os bits de controle são TON=1, TCS=1 e TGATE=0, o registrador contador TMR1 é incrementado na borda de subida de cada pulso do oscilador de baixa potência de 32 KHz até que o valor do registrador contador TMR1 seja igual ao valor prefixado no registrador PR1; depois o TMR1 é resetado (0).
- ❑ Para que o timer1 opere corretamente no modo RTC, bit de controle TSYNC tem que ser resetado. Configurando TSYNC=0 o modo assíncrono do timer1 é setado, o bit LPOSCEN (T1CON<1>) tem que ser setado para desabilitar todos os outros modos de operação excepto o modo RTC e habilitar o wake – up (acordar) do microcontrolador desde um estado SLEEP. É importante que o bit TSIDL seja resetado para permitir a operação do timer1 durante o estado IDLE.

Conexão do cristal no modo RTC

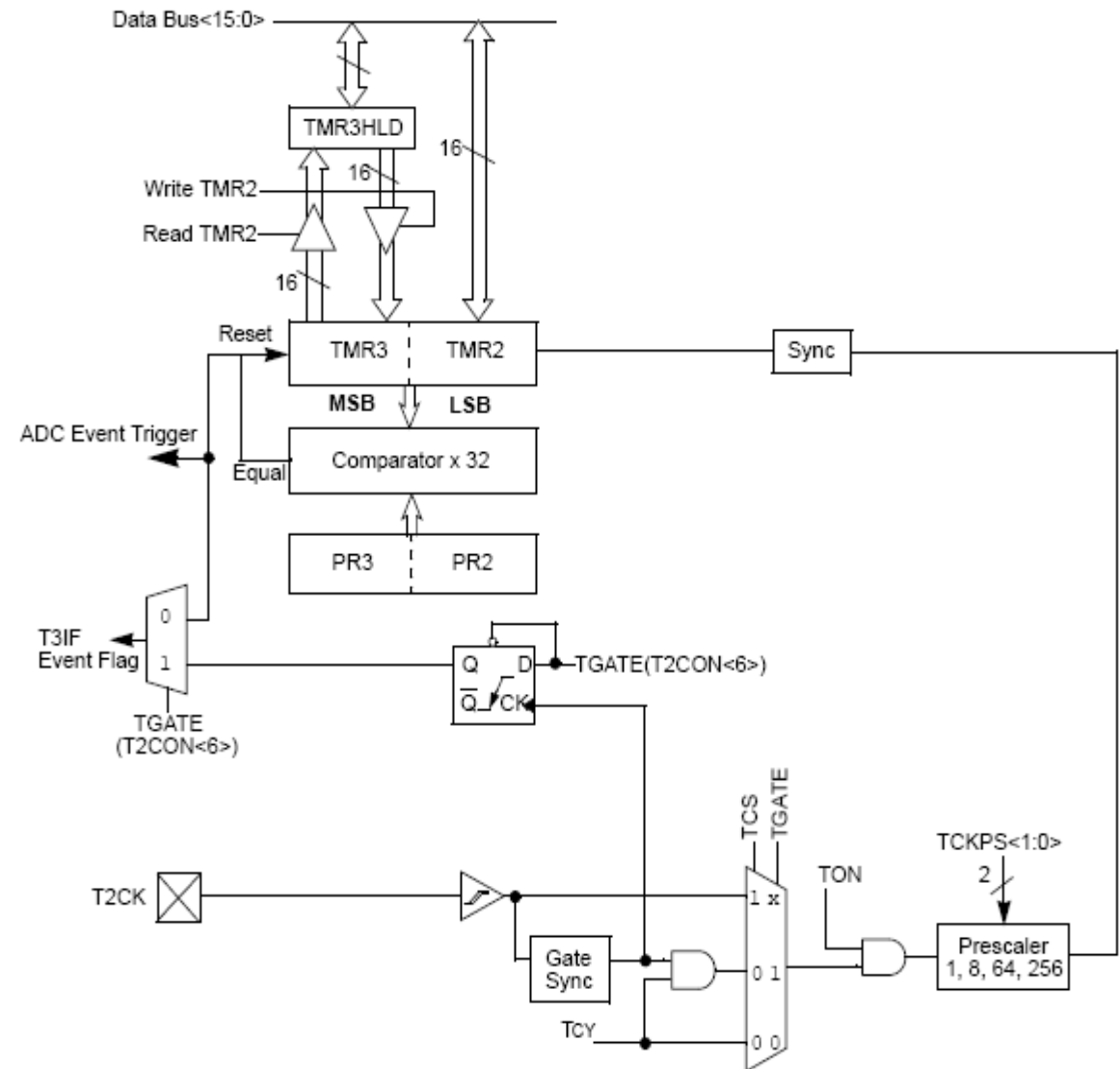


MODULO TEMPORIZADOR 2/3

(TIMER2/3)

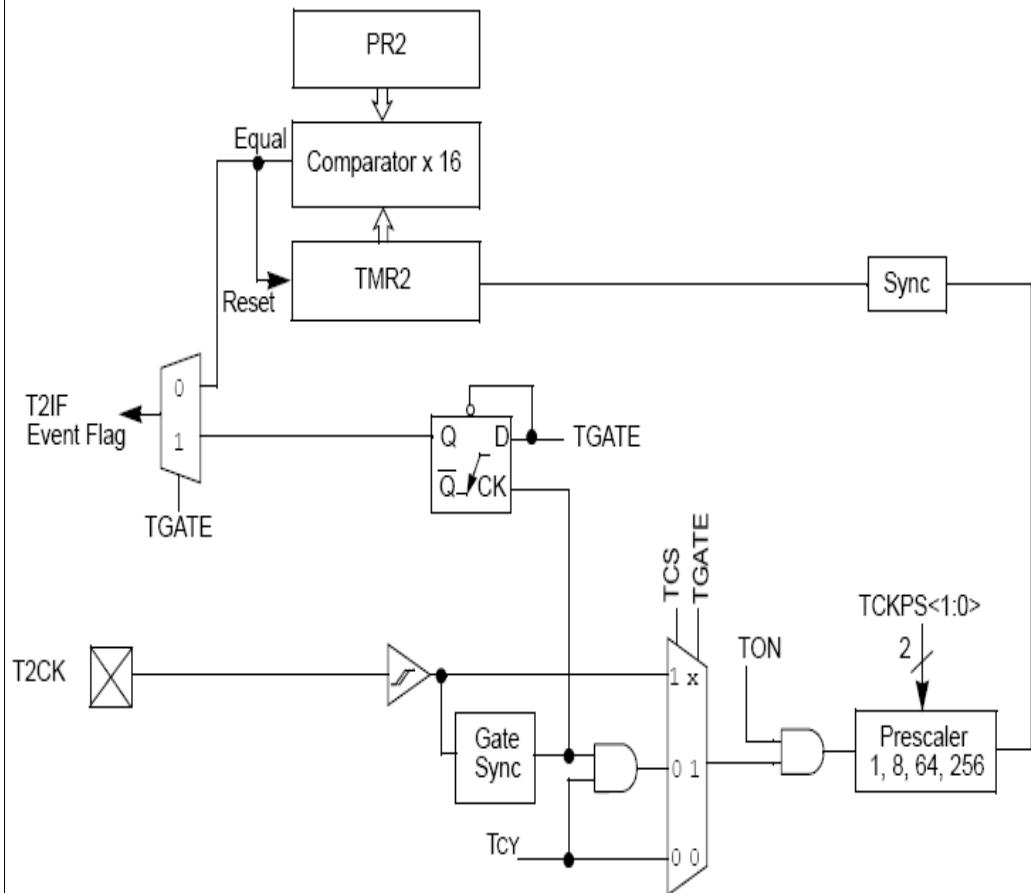
- ❑ Aqui se descreve o modulo temporizador de propósito geral de 32 bits (Timer 2/3) e modos operacionais associados.
- ❑ O timer2 é um timer tipo B (16 bits, pode ser concatenado com tipo C para formar timer de 32 bits, sincronização do clock depois do prescaler) e o timer3 é tipo C (16 bits, pode ser concatenado com tipo B para formar timer de 32 bits, tem habilidade de disparar a conversão A/D).
- ❑ O timer de 32 bits tem os seguintes modos: dois timers independentes de 16 bit (timer2 e timer3) com todos os modos de operação do timer1 (excepto o modo contador assíncrono), operação de um timer de 32 bits e de um contador síncrono de 32 bits.

32-BIT TIMER2/3 BLOCK DIAGRAM

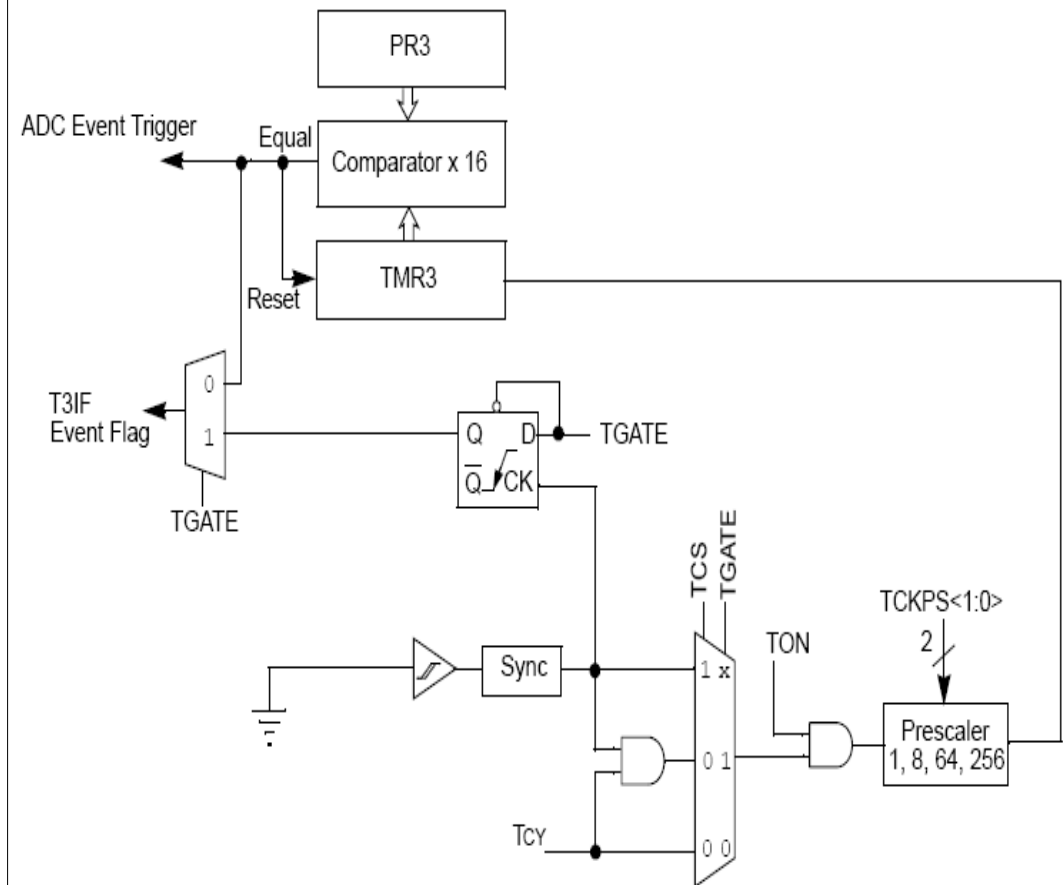


Note: Timer configuration bit, T32 (T2CON<3>), must be set to '1' for a 32-bit timer/counter operation. All control bits are respective to the T2CON register.

16-BIT TIMER2 BLOCK DIAGRAM



16-BIT TIMER3 BLOCK DIAGRAM



Note: The dsPIC30F4011/4012 devices do not have external pin inputs to Timer3. In these devices, the following modes should not be used:

1. TCS = 1.
2. TCS = 0 and TGATE = 1 (gated time accumulation).

- ❑ As seguintes características operacionais são suportadas: Disparo do evento ADC, operação timer gate, configuração de prescaler selecionável, operação do timer durante os modos IDLE e SLEEP e interrupção quando casa o valor do registrador de período de 32 bits. Estes modos de operação são determinados pela configuração apropriada dos bits dos registradores de funções especiais (SFR) de 16 bits T2CON e T3CON.
- ❑ Para a operação do timer/counter de 32 bits, o Timer2 é a palavra menos significativa e Timer3 é a palavra mais significativa do timer de 32 bits.
- ❑ Para operação do timer de 32 bits, os bits de controle T3CON são ignorados. Somente os bits de controle de T2CON são usados para configuração e controle. As entradas de clock e gate do timer2 são usadas pelo timer2/3 concatenado de 32 bits. Uma interrupção do timer de 32 bits é gerada com o flag T3IF e o correspondente bit de habilitação T3IE.
- ❑ **Modo 16 bits:** neste modo o Timer2 e Timer3 podem ser configurados como dois timers independentes de 16 bits. Cada timer pode ser setado no modo timer 16 bits ou no modo contador síncrono de 16 bits. A única diferença funcional entre o Timer2 e o Timer3 é que o Timer2 fornece sincronização da saída de clock do prescaler. Este é útil para entradas externas de clock de alta frequência.
- ❑ **Modo Timer 32 bits:** o timer é incrementado em cada ciclo de instrução até que o valor concatenado no registrador contador TMR3/TMR2 é igual ao valor do registrador de período concatenado PR3/PR2. Então o registrador de contador é resetado a 0 e uma requisição de interrupção é gerada com o bit T3IF. A leitura síncrona do registrador TMR3/TMR2 é feita lendo o TMR2 de 16 bits do Timer2 como a palavra menos significativa (Less Significant word - LSW).

- ❑ Durante a leitura do registrador TMR2 o valor do registrador TMR3 é transferido para o registrador temporário TMR3HLD. O processo da leitura de 32 bits é concluída lendo o valor da palavra mais significativa (Most Significant Word – MSW) desde o registrador TMR3HLD.
- ❑ A escrita síncrona de 32 bits é executada em dois passos inversamente à leitura. Primeiro a MSW é escrita em TMR3HLD e então a LSW é escrita em TMR2. Durante a escrita no registrador TMR2 os valores de TMR3HLD e o registrador contador TMR3 são transferidos a TMR2.
- ❑ **Exemplo:** Ligar e desligar um led na porta B ~~aproximadamente uma vez a cada dois segundos~~. O exemplo usa o timer2 e 3 concatenados sendo que o clock do timer é 256 vezes mais lento que o clock no dispositivo dsPIC. A cada 100 000 clocks do timer a rotina de interrupção é chamada e o led é desligado ou ligado.

```
#include <p30f4011.h>
//=====
_FOSC( CSW_FSCM_OFF & XT_PLL8); //A frequencia do XTAL é de 8 MHZ no kit, utilizamos PLL8.
_FWDT(WDT_OFF);
_FBORPOR(PBOR_ON & PWRT_OFF & MCLR_EN);
//=====
void __attribute__ ((interrupt, no_auto_psv)) _T3Interrupt(void)
{
    IFS0bits.T3IF = 0;
    LATBbits.LATB0 = ~LATBbits.LATB0; //Complementamos o bit.
}
//***** Programa Principal *****/
void main (void)
{
    ADPCFG = 0xFFFF; //configura a porta B (PORTB) como entradas/saídas digitais
    TRISB=0;          //a PORTB como saída
    IFS0=0;            //Flag de interrupção do timer3
    LATB=0;
    IEC0 = IEC0 | 0x0080; // bit 7 do registrador IEC0 habilita a interrupção do timer3 (IEC0bits.T3IE=1)
    PR2 = 34464; //A interrupção do período é 100 000 clocks
    PR3 = 0x0001; //Total PR3/2 = 1*65536 + 34464 ==> PR3/2=PR3*65536 + PR2
    T2CON=0x8038; //ativamos o timer2/3 e o Prescaler fica em 256 (clock interno é dividido por 256).
    while(1); //laço infinito
}
```

A SOLUÇÃO ACIMA ESTÁ CERTA ????

Registadores utilizados para trabalhar com os timers

Name SFR	Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMR1	0100	Timer1 Register															
PR1	0102	Timer1 Period Register															
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—
TMR2	0106	Timer2 Register															
TMR3HLD	0108	Timer3 Holding Register (used in 32-bit mode only)															
TMR3	010A	Timer3 Register															
PR2	010C	Timer2 Period Register															
PR3	010E	Timer3 Period Register															
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—
TMR4	0114	Timer4 Register															
TMR5HLD	0116	Timer5 Holding Register (used in 32-bit mode only)															
TMR5	0118	Timer5 Register															
PR4	011A	Timer4 Period Register															
PR5	011C	Timer5 Period Register															
T4CON	011E	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—
T5CON	0120	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—
IFS0	0084	CNIF	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF	T3IF	T2IF	OC2IF	IC2IF	T1IF	OC1IF	IC1IF	INT0IF
IFS1	0086	IC61F	IC5IF	IC4IF	IC3IF	C1IF	SPI2IF	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	IC8IF	IC7IF	INT1IF
IEC0	008C	CNIE	MI2CIE	IC2IE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE	T3IE	T2IE	OC2IE	IC2IE	T1IE	OC1IE	IC1IE	INT0IE
IEC1	008E	IC6IE	IC5IE	IC4IE	IC3IE	C1IE	SPI2IE	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	IC8IE	IC7IE	INT1IE
IPC0	0094	—	T1IP<2:0>			—	OC1IP<2:0>			—	IC1IP<2:0>			INT0IP<2:0>			
IPC1	0096	—	T3IP<2:0>			—	T2IP<2:0>			—	OC2IP<2:0>			IC2IP<2:0>			
IPC5	009E	—	INT2IP<2:0>			—	T5IP<2:0>			—	T4IP<2:0>			OC4IP<2:0>			

EXERCÍCIOS SOBRE TIMERS

- 1) Gerar no pino 1 da porta B um sinal com uma frequência de 1 KHz.
- 2) Pressionando e soltando uma tecla deve ser gerado um sinal de 1 KHz e depois de pressionar e soltar novamente a tecla gerar um sinal de 10 KHz. No pino 0 da porta B.
- 3) Gerar uma forma de onda com um período de 3 seg e outro com 4 seg nos pinos 0 e 1 da porta B respectivamente. Utilizar timer de 32 bits.
- 4) Gerar um sinal com período de 4 seg com um duty cycle de 30 %. Utilizar timer de 32 bits.
- 5) Pressionando e soltando (ativando) uma tecla gerar um sinal com período de 3 seg e duty cycle de 50%, ativando outra tecla esse mesmo sinal passa para um duty cycle de 20% e ativando mais outra tecla o D.C=80%. A saída deve ser ligada à ventoinha. Utilizar timer de 32 bits.
- 6) Uma tecla ligada ao T1CK do dsPIC servirá para fazer a contagem de quantas vezes foi pressionado e quando chegar a 10 contas, ativar o buzzer por dois segundos e voltar ao estado anterior. Cada conta deve aparecer nos displays de 7 segmentos.
- 7) Utilizando o exercício 5, o led infravermelho e os displays de 7 segmentos, fazer a contagem do número de voltas da ventoinha para cada uma das situações dadas no exercício 5.

FALTAM OS MÓDULOS TIMERS 4/5 !!!!!